

CLAIMS

1. A method for installing software components, comprising:
 - identifying components previously installed on a system;
 - identifying components to be installed on the system; and
 - identifying any potential conflicts between a previously installed component and a component to be installed.
2. The method of claim 1, wherein identifying components previously installed on a system comprises accessing a semantic model comprising relationships among previously installed components.
3. The method of claim 2, further comprising updating the semantic model with the identity of newly installed components.
4. The method of claim 1, further comprising providing a user with a plurality of options if a conflict is identified.
5. The method of claim 4, wherein a first option includes aborting the installation.
6. The method of claim 4, wherein a second option includes continuing the installation.
7. The method of claim 6, further including, upon the exercise of the second option, recording an entry in a log indicative of the conflict and of the continuation of installation.
8. The method of claim 1, further comprising:
 - initiating a removal of a component from the system; and
 - identifying remaining components which depend on the component to be removed.

9. The method of claim 8, further comprising providing a user with a plurality of options if a dependent remaining component is identified.
10. The method of claim 9, wherein a first option includes aborting the removal.
11. The method of claim 9, wherein a second option includes continuing the removal.
12. The method of claim 8, further comprising:
 - identifying a first component previously installed on the system which is dependent upon a removed component; and
 - indicating the identity of a second component upon which the first component depends.
13. The method of claim 12, further comprising:
 - installing the second component upon which the first component depends;
 - and
 - creating a dependency link between the first and second components.
14. A system for installing software components, comprising:
 - means for identifying components previously installed on a system;
 - means for identifying components to be installed on the system; and
 - means for identifying any potential conflicts between a previously installed component and a component to be installed.
15. The system of claim 14, wherein the means for identifying components to be added to the system comprises means for accessing a semantic model comprising references among the components to be installed.
16. The system of claim 15, further comprising means for loading an installation package including the semantic model.

17. The system of claim 14, further comprising a data structure comprising references among the components to be installed.
18. The system of claim 17, further comprising means for accessing the data structure.
19. The system of claim 14, further comprising means for installing the components across a plurality of enterprise resources.
20. A data structure associated with a software component installation package for identifying component incompatibilities, comprising:
an entry for each component previously installed on a system; and
references associated with each previously component identifying any conflicting component to be installed;
whereby an alert is generated if an attempt is made to install a conflicting component.
21. A computer program product of a computer readable medium usable with a programmable computer, the computer program product having computer-readable code embodied therein for installing software components, the computer-readable code comprising instructions for:
identifying components previously installed on a system;
identifying components to be installed on the system; and
identifying any potential conflicts between a previously installed component and a component to be installed.
22. The computer program product of claim 21, wherein the instructions for identifying components previously installed on a system comprise instructions for accessing a semantic model comprising relationships among previously installed components.

23. The computer program product of claim 22, further comprising instructions for updating the semantic model with the identity of newly installed components.
24. The computer program product of claim 21, further comprising instructions for providing a user with a plurality of options if a conflict is identified.
25. The computer program product of claim 24, wherein a first option includes aborting the installation.
26. The computer program product of claim 24, wherein a second option includes continuing the installation.
27. The computer program product of claim 26, further including instructions for, upon the exercise of the second option, recording an entry in a log indicative of the conflict and of the continuation of installation.
28. The computer program product of claim 21, further comprising instructions for:
initiating a removal of a component from the system; and
identifying remaining components which depend on the component to be removed.
29. The computer program product of claim 28, further comprising instructions for providing a user with a plurality of options if a dependent remaining component is identified.
30. The computer program product of claim 29, wherein a first option includes aborting the removal.
31. The computer program product of claim 29, wherein a second option includes continuing the removal.

32. The computer program product of claim 28, further comprising instructions for:
identifying a first component previously installed on the system which is dependent upon a removed component; and
indicating the identity of a second component upon which the first component depends.
33. The computer program product of claim 32, further comprising instructions for:
installing the second component upon which the first component depends;
and
creating a dependency link between the first and second components.
34. A method for installing software components, comprising:
loading an installation package, the installation package including a component compatibility data structure;
searching a target to which components are to be installed to identify installed components;
accessing the component compatibility data structure, the component compatibility data structure comprising, for each component A to be installed, a reference to any installed component with which the component A may conflict;
and
determining whether a conflict is detected.
35. The method of claim 34, further comprising notifying a user of the conflict.
36. The method of claim 34, further comprising aborting the installation if a conflict is detected.
37. The method of claim 34, further comprising ignoring a detected conflict and continuing the installation.
38. The method of claim 37, further comprising entering a note in a log of the conflict.

39. The method of claim 34, further comprising:
initiating the removal of an installed component;
accessing the component compatability data structure; and
identifying a conflict if the installed component is removed.
40. The method of claim 34, further comprising:
initiating an installation of a component B;
searching a target to which the component B is to be installed to identify
installed components;
accessing the component compatability data structure; and
determining if all of the components required by the component B are
installed.